

## CLAIMS

What is claimed is:

1. A multi-thread command processing system comprising:  
a reservation station having a plurality of command threads stored therein;  
an arbiter operably coupled to the reservation station such that the arbiter retrieves a first command thread of the plurality of command threads stored therein; and  
a command processing engine operably coupled to receive the first command thread from the arbiter such that the command processor performs at least one processing command from the first command thread and thereupon updates the first command thread in the reservation station.
2. The command processing system of claim 1 wherein each of the plurality of command threads include a done flag which is activated after all commands with the command thread have been executed by the command processing engine.
3. The command processing system of claim 1 wherein the reservation station is a memory device and the arbiter retrieves the first command thread based on a priority scheme.
4. The command processing system of claim 1 further comprising:  
the arbiter retrieves a second command thread of the plurality of command threads stored therein; and  
the command processing receiving the second command thread wherein the first command thread and the second command thread may be interleaved.
5. The command processing system of claim 1 wherein the command processing engine is a texture processing engine.
6. The command processing system of claim 1 further comprising:

an arithmetic logic unit (ALU) operably coupled to the arbiter such that the arbiter is capable of providing at least one of the plurality of command threads to the ALU.

7. The command processing system of claim 1 wherein the reservation station is at least one of: a vertex reservation station and a pixel reservation station.

8. A multi-thread graphics processing system comprising:
- a first reservation station having a plurality of first command threads stored therein;
  - a second reservation station having a plurality of graphic command threads stored therein;
  - an arbiter coupled to the first reservation station and the second reservation station such that the arbiter retrieves a selected command thread from one of the plurality of command threads; and
  - a graphics processing engine operably coupled to the arbiter such that the selected command thread is received and processed by the graphics processing engine.
9. The multi-thread graphics processing system of claim 8 further comprising:
- the graphics processing engine being operably coupled to the first reservation station and the second reservation station such that upon processing the selected command thread, the selected command thread is updated in the first reservation station or the second reservation station from wherein it was previously retrieved.
10. The multi-thread graphics processing system of claim 8 wherein the first reservation station is a memory device and the second reservation station is a FIFO memory device.
11. The multi-thread graphics processing system of claim 8 further comprising:
- the arbiter retrieves a second selected command thread from one of the plurality of command threads; and
  - the command processing engine receiving the second selected command thread such that the first selected command thread and the second selected command thread may be interleaved.

12. The multi-thread graphics processing system of claim 8 wherein the graphics processing engine is a texture processing engine, the system further comprising:

an arithmetic logic unit (ALU) operably coupled to the arbiter such that the arbiter is capable of providing at least one of the first selected command thread and the second selected command thread to the ALU.

13. The multi-thread graphics processing system of claim 8 wherein the first reservation station is a vertex reservation station and the second reservation station is a pixel reservation station.

14. A graphics processing system comprising:
- a pixel reservation station having a plurality of pixel command threads stored therein;
  - a vector reservation station having a plurality of vector command threads stored therein;
  - an arbiter coupled to the pixel reservation station and the vector reservation station;
  - an arithmetic logic unit (ALU) operably coupled to the arbiter; and
  - a texture engine operably coupled to the arbiter wherein the arbiter retrieves a first selected command thread from one of the plurality of pixel command threads and the plurality of vector command threads and the arbiter thereupon provides the first selected command thread to at least one of: the ALU and the texture engine.
15. The graphics processing system of claim 14 further comprising:
- the arbiter retrieves a second selected command thread from one of the plurality of pixel command threads and the plurality of vector command threads and thereupon provides the second selected command thread to at least one of the ALU and the texture engine.
16. The graphics processing system of claim 15 wherein when the first selected command thread and the second selected command thread are both provided to the ALU, the first selected command thread may be interleaved with the second selected command thread.
17. The graphics processing system of claim 14 wherein the ALU and the texture engine are operably coupled to the pixel reservation station and the vector reservation station such that the first selected command thread may be provided back from wherein it was previously retrieved.
18. The graphics processing system of claim 14 wherein the pixel reservation station includes a first pixel memory device storing a plurality of pixel state bits and a second pixel memory device storing a plurality of pixel status bits and the vector reservation station includes a

first vector memory device storing a plurality of vector state bits and a second vector memory device storing a plurality of vector status bits.

19. The graphics processing system of claim 14 wherein the pixel command threads and the vector command threads each include a done flag, wherein when the done flag is activated in the pixel command thread, the pixel command thread is provided a rendering backend and when the done flag is activated in the vertex command thread, the vertex command thread is provided to a scan converter.

20. A method for multi-thread graphics processing comprising:  
retrieving a selected command thread from a plurality of command threads;  
providing the selected command thread to a graphics processing engine;  
performing a command in response to the selected command thread; and  
writing the selected command thread to a first reservation station if the selected command thread is one of a plurality of first command threads and the to a second reservation station if the selected command thread is one of a plurality of second command threads.
21. The method of claim 20 further comprising:  
retrieving a second selected command thread from the plurality of command threads;  
providing the second command thread the graphics processing engine;  
prior to writing the selected command thread to either the first station or the second reservation station, interleaving the selected command thread and the second selected command thread;  
performing a second command in response to the selected command thread; and  
writing the second selected command thread to a first reservation station if the selected command thread is one of the plurality of first command threads and the to a second reservation station if the selected command thread is one of the plurality of second command threads.
22. The method of claim 20 further comprising:  
prior to providing the selected command thread to the graphics processing engine,  
providing the selected thread command to at least one of: the graphics processing engine or an arithmetic logic unit (ALU); and

if the second selected command thread is provided to the ALU:

performing a command in response to the selected command thread; and

writing the selected command thread to the first reservation station if the selected command thread is one of the plurality of first command threads and the to the second reservation station if the selected command thread is one of the plurality of second command threads.

23. The method of claim 20 wherein the graphics processing engine is a texture engine.

24. The method of claim 20 further comprising:

when all of the commands with the selected command thread have been executed, setting a done flag.

25. The method of claim 24 wherein the first plurality of command threads are pixel command threads and the second plurality of command threads are vertex command threads, the method further comprising:

when the done flag is set on the pixel command thread, providing the pixel command thread to a render backend; and

when the done flag is set on the vector command thread, providing the vector command thread to a scan converter.